Supreme Court, U. S.
FILED
OCT 16 1975

IN THE

SUPREME COURT OF THE UNITED STATES

October Term, 1975

No. 74-1033

C. MARSHALL DANN,
COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner

v.

THOMAS R. JOHNSTON,
Respondent

On Writ of Certiorari to the United States Court
of Customs and Patent Appeals

**Brief Amicus Curiae for the
Association of Data Processing Service Organizations,
Software Industry Association
(ADAPSO/SIA)**

DAVID COHEN
Attorney for Amicus

ADAPSO/SIA
1420 Walnut Street
Philadelphia, Pa. 19102

THE LEGAL INTELLIGENCER, 66 NORTH JUNIPER STREET, PHILA., PA. 19107 (215) 561-4850

TABLE OF CONTENTS

	Page
Questions Presented	2
Interest of Amicus Curiae	2
Nature of the Industry	3
A. The Absence of Competition in Software	3
B. The Resulting Poor-Quality Software	6
C. The Small Company Potential for Competition	8
D. The Importance of Patent Incentives	8
Summary of Argument	12
I. The Writ of Certiorari Should Be Dismissed	15
A. The record in this case is insufficient to permit consideration of the questions presented by the Government	15
B. This Court is without jurisdiction to reverse findings of the Patent Board of Appeals favorable to respondent	17
C. The basic question for which review was sought in this case—whether novel and unobvious machines constructed by programming a general-purpose digital computer are patentable—cannot be resolved by a decision in this case and Petitioner so concedes	18
II. Special-Purpose Computers Built by Programming a General-Purpose Computer Are Machines, Hence Patentable Subject Matter	20
A. The technology of computers and computer programs	20
B. A special-purpose computer built by programming a general-purpose computer is a different machine than the general-purpose computer with its own unique circuit structure	22

Page

III. Software Is Not An "Abstract Idea" 26
A. The engineering of software 26
B. The invention process in software 29
IV. Copyrights Cannot Protect Machine Inventions; Patents, Which Can, Are Needed For Software .. 30
Conclusion 32

TABLE OF CITATIONS

Cases:

Brenner v. Manson, 383 U.S. 519 ..... 16, 18
Glidden v. Zdanok, 370 U.S. 530 ..... 17, 18, 19
Gottschalk v. Benson, 409 U.S. 63 25
Graham v. John Deere Co., 383 U.S. 1 31
In re Bernhart, 417 F.2d 1935 (1969) 23, 25
In re Foster, 438 F.2d 1011 (1971) 23
In re Mahoney, 421 F.2d 742 (1970) 23
In re Melott, 331 F.2d 887 17
In re Musgrave, 431 F.2d 882 (1970) 23
In re Prater, 415 F.2d 1393 (1969) 23, 25
Johnson v. Massachusetts, 390 U.S. 511 15
Massachusetts v. Painten, 389 U.S. 560 15
U.S. v. IBM, 69 Civ. 200, S.D.N.Y. 3, 4, 5, 8, 11

Statutes:

28 U.S.C. §1256 17
35 U.S.C. §7 ..... 15, 16
35 U.S.C. §101 19

Statutes:

Page

35 U.S.C. §103 ..... 12, 19
35 U.S.C. §112 25
35 U.S.C. §141 ..... 12, 15, 17
35 U.S.C. §144 ..... 15, 19
42 U.S.C. §281(a) 19

Miscellaneous Authorities:

Berteau, "IBM and the Structure of the Industry—The Software Monopoly," 21 *Datamation* 111, Oct., 1975 3

Brenner, Commissioner of Patents, "Guidelines to Examination of Programs," 829 *Official Gazette of the Patent Office* 441, 442, July 6, 1966 ..... 9, 22

Brenner, Commissioner of Patents, "Revised Guidelines to Examination of Programs," 855 *Official Gazette of the Patent Office* 829, 830, Oct. 22, 1968 9

Brenner, Commissioner of Patents, "The Future of Computer Programs in the U.S. Patent Office," *Law of Software* 1968 ..... 5, 9, 10

Brooks Subcommittee Report on H.R. 4845, Aug. 17, 1965, 89th Cong., 1st Sess. House Rep. No. 802 "Domination of ADP Industry by IBM," 3

"Composite Design: The Design of Modular Programs," Report of IBM Poughkeepsie Labs, Jan., 1973 ... 7, 27

Computerworld, Nov. 20, 1974 6

Computerworld, Aug. 6, 1975 6

Davis, Dr. Ruth M., Statement before the Government Activities Subcommittee, House Committee on Government Operations, May 16, 1972 6

TABLE OF CITATIONS—(*Continued*)

Miscellaneous Authorities: Page

"EDP—A 20 Year Ripoff," *Infosystems*, 11/74 ..... 7, 8

"Hierarchies—Software Engineering Design," *Software Engineering, International Computer State of the Art Report*, 1972 26

Jacobs, "Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyrights and Patents," *Rutgers Journal of Computers and Law*, Vol. 1970, No. 8 ..... 23, 26, 27

"Management Concepts for Top-Down Structured Programming," IBM Technical Report, Feb., 1973 .. 28

National Bureau of Standards Experimental Technology Inventives Program, Preliminary Program Description, Fiscal Year 1973, Budget Submission 8

2 *New Encyclopaedia Britannica: Macropaedia* 505-506 (15th ed. 1974) "Automation" 20

"155, 165 Owners Angry With IBM," *Datamation*, August, 1974 7

Rich, "Change," 2 *APLA Quarterly Journal* 214, American Patent Law Association, Fall, 1974 11

" 'Software Dinosaur' Slows Growth" Gorchow, 7 *Software Digest* 2, May 22, 1975 7

"Software Engineering," Report on a World Conference Sponsored by the NATO Science Committee, Garmisch, Germany, Jan., 1969 29

Turn, *Computers in the 1980's*, Colum. Univ. Press 21, 1974 29

Weinberg, *Psychology of Programming*, Van Nostrand Press 26

WIPO—AGCP/NGO/9, July 5, 1974 31

WIPO—ACGP/NGO/II/11, July 14, 1975 32

IN THE

SUPREME COURT OF THE UNITED STATES

October Term, 1975

No. 74-1033

C. MARSHALL DANN,
COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner

v.

THOMAS R. JOHNSTON,
Respondent

On Writ of Certiorari to the
United States Court of Customs and Patent Appeals

BRIEF AMICUS CURIAE FOR THE ASSOCIATION OF DATA PROCESSING SERVICE ORGANIZATIONS, SOFTWARE INDUSTRY ASSOCIATION (ADAPSO/SIA)

The Association of Data Processing Service Organizations, Software Industry Association, files this brief *amicus curiae* in opposition to Petitioner, and also files letters of consent of the parties in accordance with the Rules.

QUESTIONS PRESENTED

1. Whether the writ of *certiorari* should be dismissed in a case in which the record is inadequate, there is a jurisdictional barrier to the consideration of a central issue, and the principal question for which *certiorari* was requested cannot be resolved?

2. Whether an otherwise patentable special-purpose machine becomes unpatentable solely because it is constructed by programming a general-purpose computer?

INTEREST OF AMICUS CURIAE

Association of Data Processing Service Organizations, Software Industry Association (hereinafter ADAPSO/SIA) is a nonprofit trade association of 62 companies in the computer field,[1] approximately 70% of whom concentrate their attention in the software products industry, and the remainder largely in the furnishing of varied software services to computer users and to hardware manufacturers.

ADAPSO/SIA presents its views to the Court because an inaccurate description of software technology and the software products industry has been set forth by Petitioner's Brief and the supporting brief of the hardware manufacturers' association, CBEMA.

ADAPSO/SIA companies are not "idea or think tanks." They are companies engaged in building and marketing machines for which profit-conscious companies will and do expend money. ADAPSO/SIA's members believe they are entitled to the same patent protection for their innovative products as that granted the hardware manufacturers.

---

1. ADAPSO/SIA (ADAPSO/Software Industry Association) was formed by merger in June 1972 of the Software Products and Service Section of ADAPSO and the Association of Independent Software Companies (AISC), the latter comprising 8 companies and the former 25 companies.

NATURE OF THE INDUSTRY

A. *The Absence of Competition in Software.*

Whereas ADAPSO/SIA is an organization of relatively small companies, with no one company exerting dominance over all the others, CBEMA is an organization of relatively large companies, operating in a computer hardware industry "shaped and dominated by IBM."[2] IBM's dominance is such that its 70% or better share of the computer hardware market is nearly ten times that of its nearest competitor, Sperry-Rand.[3] Other giant companies such as Honeywell, Burroughs, Control Data and National Cash Register are compelled to share the remaining portion of the market; General Electric and RCA found themselves forced to withdraw entirely from competition. Thus, it is IBM's voice and actions which have and which do dominate the computer industry.

In this case the Government remains silent while CBEMA distorts the history of the computer industry to assert that "the computer industry has developed in an atmosphere of free interchange of computer programs. . . ."[4] How "free" the atmosphere has been in the computer industry is best shown by the Government's pretrial brief in *United States of America v. International Business Machines Corporation*.[5]

---

2. Government's Pretrial Brief, *U.S. v. IBM*, 69 Civ. 200, S.D.N.Y. (hereinafter Pretrial Br.)

3. See Report, Brooks Subcommittee on H.R. 4845, Aug. 17, 1965, 89th Cong. 1st. Sess. (House Rep. No. 802) "Domination of ADP Industry by IBM," p. 11.

4. CBEMA Brief 10-12.

5. Pretrial Br. *supra*.

See also Berteau, "IBM and the Structure of the Industry, The Software Monopoly, 21 *Datamation* 111, October, 1975:

> We have seen hardware technology continue to advance at a rapid pace but the systems architecture seems to stay the same. Have you wondered why? The answer is *software*. As long as the structure of system software is fixed, innovation in system architecture is limited. . . .

The Government in its antitrust action against IBM notes the development of the computer industry and finds that in "the earliest days of the development of the computer industry, there was nothing that could be identified as a software industry."[6] This was in part because users design some of their own software and because the "computer systems manufacturers, following IBM's lead, provided [free] much of the software [that] users required."[7]

It was not until the late 1960's that "the independent software *products* market emerged, pioneered by the firm Applied Data Research, Inc.,"[8] one of the *amici* herein. As the Government notes, "in light of the fact that IBM provided its customers with their software requirements as part of the IBM 'total systems solution,' independent software product firms had to market their offerings against products for which there was no identifiable price. As a result, only those companies which could offer products not available 'free' from IBM or which could demonstrate the technological superiority of their products in comparison to IBM's had any chance of survival."[9] The result was that

Note 5—*Continued*

> The most significant reason for this situation probably lies in the history of software building by manufacturers. It's just not in the manufacturers' interest to produce optimum system software. It might result in the manufacturer selling less iron. . . .
>
> The solution to the software building problem brings us to a real issue facing the industry today—monopoly and restraint of trade in software. We have a small but significant competitive market in applications packages. There is very little competition in sorts and compilers, almost no competition in data communications access methods, and no competition in operating systems. And unless the U.S. antitrust lawyers understand the significance of this, we'll have even less competition in the future.

The author is identified by the magazine as "a holder of one of the early software patents."

6. Pretrial Br. 28.
7. Pretrial Br. 28.
8. Pretrial Br. 28.
9. Pretrial Br. 29.

the independent software product firms lacked "a realistic opportunity to compete."[10] That this was not an innocent case of cause and effect was shown by IBM internal memoranda which revealed that "the value of this exclusionary effect derived from bundled pricing [so-called free software][11] was fully appreciated by IBM . . . The effect of IBM's ability to wield such exclusionary power has been felt most acutely . . . by those firms seeking to market software products."[12]

The Government has also noted that "IBM internal memoranda . . . reflect that from as early as 1965, IBM was aware of the marketing benefits derived from bundled pricing and the anticompetitive effect it had on software companies."[13] The net effect was that "bundled pricing enabled IBM to exclude competition from independent software vendors."[14]

The Government concludes that its study of IBM documents "illustrate without ambiguity that its bundled pricing marketing approach was a practice known by IBM to be legally indefensible. Nevertheless, IBM consciously continued bundled pricing because of the competitive leverage it afforded IBM in maintaining its monopoly position."[15]

Software monopolies are also maintained by other hardware manufacturers through the use of bundling. A small software company has sued Sperry Rand, charging it with refusing to place a price on its bundled software and thereby making it impossible for the small company to sell

10. Pretrial Br. 29.
11. The Commissioner of Patents punctured this boasting of free software in 1968: "Today, the only significant free contribution of software to the public comes from the universities and other organizations dedicated to research rather than profit." Brenner, Commissioner of Patents, "The Future of Computer Programs in the U.S. Patent Office," *Law of Software*, 1968, pp. B-1, 3.
12. Pretrial Br. 201.
13. Pretrial Br. 205.
14. *Ibid.*, 207.
15. *Ibid.*, 236.

its software in competition.[16] Honeywell in bundling its software and hardware asserts that the software is "free." But when the hardware buyer attempts to transfer that software along with the hardware to another company, Honeywell claims that it owns the software and will control who can have possession.[17]

B. *The Resulting Poor-Quality Software.*

The CBEMA brief also boasts of "The tremendous growth of the computer software industry and the vast number of breakthroughs in the field of programming, which have occurred in the last two decades" (CBEMA Br. 12). How different the facts really are in the computer software field is made clear by the testimony of Dr. Ruth M. Davis, Director of the Center for Computer Services and Technology of the National Bureau of Standards[18] characterizing the present condition of software as "poor-quality software," a condition which she described as the "crucial computer problem" confronting the nation today. Speaking on the "seriousness of the problems besetting both the consumers of computer products and services and the suppliers of these products and services" Dr. Davis stated:

> The crippling problems are lining up in the software rather than in the hardware field. In particular, they fall under what we call the application-independent problem category . . . these are the software problems which are common to many applications.

That the condition described by Dr. Davis in 1972 still holds true is seen by her warning at the CBEMA Spring meeting, May 29, 1974 that "the difficulty and cost of

16. *Computerworld*, November 20, 1974, p. 45.

17. *Computerworld*, August 6, 1975, p. 1.

18. Testimony given before the Government Activity Subcommittee, House Committee on Government Operations, May 16, 1972.

developing and maintaining quality software have become the single largest consideration in computer utilization and composes a serious limitation on the cost-effective use of computers."

Even IBM acknowledges that "the biggest problem facing programming today is the extreme difficulty and cost encountered in creating and maintaining large programming systems."[19] At the National Computer Conference held early this year it was admitted by a leading hardware manufacturer that the "software dinosaur" has not "kept pace with the technical needs of the industry."[20] As a result of the software users being at the mercy of the hardware manufacturers, the users are forced to make do with the "free," poor and technically inferior software.[21]

The reason for the poor state of the development of software lies in the shackles IBM has placed upon free competition. History shows that it has been small technological companies and independent inventors, and not the vast financial combines such as IBM, which have been responsible for much of the innovation of this age. A recent govern-

19. The IBM report from their Poughkeepsie Laboratory — "Composite Design: The Design of Modular Programs" Jan., 1973, p. 3.

20. " 'Software Dinosaur' Slows Growth," Univac Vice President Neil Gorchow, Sperry Rand, *Software Digest*, Vol. 7, No. 20, May 22, 1975, p. 2.

21. "155, 165 Owners Angry With IBM," *Datamation*, August, 1974, p. 76. The "free" software of the hardware manufacturers has been described by a professional for 24 years in electronic data processing (EDP): ". . .Computer systems are in many cases defective. They are primarily defective in the computer programs that come with the hardware — the body of programs referred to as the 'Operating System' . . . a sort of executive in managing the work of the many parts of the computer system, . . .[the] housekeeping chores . . . [A] very big set of programs [that] is often inefficient, overcomplicated and unreliable . . . [t]he outrageous fact remains that the Operating System is often an imperfect and incomplete product which typically contains errors, omissions and inconsistencies. ("EDP — A 20 Year Ripoff," INFOSYSTEMS, 11/74, pp. 26, 27, 29.)

ment report has found that "studies of the sources of invention and innovation have shown that independent inventors and small technologically based companies are responsible for a remarkably high percentage of the important inventions and innovation of this century.[22]

C. *The Small Company Potential For Competition.*

The small, mostly "young and struggling", companies who "surely possess excellent technical innovative capabilities" and "the principal technical capability . . . to work . . . on our critical computer problems"[23] have been wrongfully denied the incentive of the patent system, which has been assured to the hardware manufacturers for products they sell. The patent incentive is both a form of recognition and a major aid in obtaining the necessary financing without which these companies face the prospect of shriveling and dying. It is no accident that the monopolistic and exclusionary practices of IBM have permitted only a small number of companies in the software products field to achieve a relative degree of success. As the Government states in its pretrial brief "operating systems continue to be part of IBM's bundled price structure. These continue to be bundled for the marketing leverage they afford IBM in terms of increasing the sale of IBM computer equipment and forestalling the growth of independent software companies."[24]

D. *The Importance of Patent Incentives.*

A patent on a software program creates an economic toehold in the monopoly-dominated computer field. The software products company characterized by the Government as the pioneer in the independent software products

22. National Bureau of Standards Experimental Technology Inventives Program, Preliminary Program Description, Fiscal Year 1973 Budget Submission, p. 9.

23. Davis, Committee Testimony, p. 15, *supra*.

24. Pretrial Br. 226.

market, Applied Data Research, Inc.,[25] stands as proof of the importance of software patents.[26]

To a large financial giant, the economic value of a patent may not loom large; to the small software companies upon which the future of the development of quality software depends, the financial value of the patent may spell the difference between life and death. To banks and financial institutions the existence of a patent or even the potentiality of obtaining one may well be a decisive factor in determining whether a loan should be granted. To prospective investors a patent or the possibility of obtaining one may be the principal element in the decision whether to invest.

Making clear that patents may be available for inventions in software would unleash enormous innovative talent. It would have the direct opposite effect forecast by IBM and the other hardware manufacturers—it would enable competition with those companies and provide the needed incentive to stimulate innovation.

Patent protection would encourage the free dissemination of program innovations which is so necessary to stimulate technological progress. Disclosure is the statutory price paid by the inventor for patent protection. Without such protection inventors principally have to rely on the law of trade secrets in order to secure the economic value of their inventions, thereby excluding from the world-at-large disclosure of the invention.

When the Patent Office for its own convenience[27] urges that inventors rely on trade secrets in lieu of patents,

25. Pretrial Br. 28.

26. *See* ADR *Amicus* Brief filed herein.

27. In 1966 the Patent Office issued Guidelines which based upon the facts of the technology, provided for the patentability of computer program inventions "Guidelines to Examination of Programs," 829 *Official Gazette*, 441, 442, July 6, 1966. In 1968, the Patent Office issued a set of Revised Guidelines, "Guidelines to Examination of Programs," 855 *Official Gazette*, 829, 830, October 22, 1968, based upon "policy" (*i.e.*, not legal reasons) limiting

it is ignoring the mandate of the Patent Laws, and when it urges copyrights as another potential alternative, it is denying the inventiveness of an entire technology, thus abdicating its responsibility under the Patent Laws to grant patents for machines that "advance the Progress of the useful Arts."

Such computer programs as are freely disclosed in the trade literature of the computer industry have little economic value or technological worth. All that this kind of "free exchange" has produced for the computer industry has been "crippling problems" and "poor quality software." The real strides forward in computer development occurred in hardware, an area fully protected by the patent laws. In a competitive economy, that which is given away for nothing tends to have the same value. Non-profit companies are non-existent in the computer industry. CBEMA favors its version of "free interchange of programs" because that has been one of its major techniques in excluding competition and in monopolizing the computer industry. It is profits and not "free interchange of programs" or "innovative software that CBEMA seeks.

The economic forces opposed to patent protection were clearly identified by Judge Rich in his dissenting opinion below. As he stated it:

> On the one side . . . against patenting programs or software [were] collective forces of major hardware (*i.e.*, computer) manufacturers and their representative associations who, for economic reasons, did not want patents granted *on programs* for their machines. (Emphasis in the original.)

This portion of Judge Rich's "dissent" (a dissent whose purpose was to provoke this Supreme Court review to

---

Note 27—*Continued*

severely the grant of patents in this area on grounds of administrative convenience, Brenner, *Law of Software*, *supra*, B-12.

obtain a ruling providing for software patents)[28] is ignored by the Government as is the fact that Judge Rich reaffirmed his view as to the soundness of his previous opinions on computer technology.

When one knows that every action taken by IBM has been motivated by its monopolistic and exclusionary desires, one understands why IBM has led the legal fight against patents for software. It is not to strengthen the computer industry but only to strengthen IBM's stranglehold upon it that IBM postulates its legal theories in opposition to patents. Strange indeed is it that the Government despite its firsthand knowledge of IBM's efforts aimed at "forestalling the growth of independent software companies"[29] joins hands herein with IBM in furtherance of IBM's monopolistic endeavors.

To arbitrarily deny patent coverage as urged herein by the Patent Office is to denigrate software technology and to cast a chilling effect upon inventiveness.

---

28. Rich, "Change," 2 *APLA Quarterly Journal* 214, American Patent Law Association, Fall, 1974): "I do not see how the Patent Office can avoid at least trying to get certiorari granted in *Johnston*, as I implied in my dissenting opinion, in which, by the way, I do not abandon my personal beliefs about what §101 means, as expressed in the C.C.P.A. *Benson* opinion, ..."

29. Pretrial Br. 226.

SUMMARY OF ARGUMENT

The writ of *certiorari* should be dismissed as improvidently granted, because of the inadequate record, the jurisdictional barrier to the reconsideration of a central factual issue, and the inappropriateness of the principal question for which *certiorari* was granted. To consider what the Patent Commissioner presents as the key issues in this case would require this Court to undertake a thorough inquiry into the complex technological issues of special-purpose computer construction, but the record does not provide the factual basis which would enable the Court to do so. The Supreme Court is not the appropriate forum for initial fact finding.

The most basic technological issue — the Board of Appeals' finding that programming a general-purpose computer restructures the computer into a different machine— is not jurisdictionally subject to review because 35 U.S.C. §141, *et seq*. bind the Commissioner of Patents by Board determinations. Since the Board is the *alter ego* of the commissioner, his attempt to appeal its finding is in essence a nonjusticiable intra-agency dispute. Moreover, the C.C.P.A. and the Board have considered the basic technological fact issue, each has special credentials of technological expertise and both have found in favor of the restructuring principle.

The broad question for which *certiorari* was granted— whether innovative machines which can be embodied in computer programs are patentable subject matter— is inappropriate for judicial resolution in this case because patentability depends on the structural nature of the specific invention not which of alternative types of construction was employed. The petitioner has conceded that the Court cannot formulate a rule of general applicability in its decision of this case and that the only other question presented—obviousness under 35 U.S.C. §103—is not worthy of this Court's attention.

The petitioner's main legal arguments depend on a mischaracterization of computer program technology. An

accurate picture of this technology reveals that a programmed general purpose computer is a different machine than the unprogrammed machine. The designer of software, like any engineer, builds machine structures, not methods of doing business; he develops machine solutions for machine problems. Computer innovations embodied with stored computer programs are legally entitled to the same patent protection as innovations embodied in wire-connected electronic circuits, for technologically "hardware" and "software" are equivalent.

The *facts* that a programmed computer (what respondent defines as his invention) is a machine and the innovative aspect of it is a machine structure cannot be overcome by petitioner's *legal* arguments. A machine is not a method of doing business. An innovative machine structure is not the mere embodiment of an abstract idea. Machines are patentable subject matter and respondent's only argument against software being patentable subject matter is the unsupported statement—appealing perhaps to a layman but universally rejected by the experts—that a program is not a machine but merely an idea, "new use," or "method of doing business."

Patent protection for software is vital to the health of the data processing industry. In a field dominated by IBM, the growth of the small independent software company has been stifled by the bundling practices of the hardware manufacturers which continue to this day in the systems programs area. Only through the availability of protection for their products can the software producers hope to bring competition to an industry that sorely needs it.

The absence of patent protection has forced reliance on trade secrecy and has had the negative effect of deterring disclosure of innovations. The availability of patents, with their public disclosure requirement, would "promote the progress of the useful Arts," in this most sophisticated, pervasive and vital area of technology.

Copyright protection is inadequate and inappropriate to protect innovative machine structures. Copyright of a

particular set of computer language statements does not protect the novel machine structure that is the core of any software invention. The Constitution establishes innovation as the prerequisite for a grant of exclusive rights to machine structures; attempts to protect them via copyright or registration run counter to this principle.

At stake in this case is equal patent treatment for the products of small software companies. The myth perpetuated by the Government and CBEMA — that computer-program embodied inventions are not machine inventions —has been exploded; there is now no justification for the denial of patent protection to software-embodied machines.

I. THE WRIT OF CERTIORARI SHOULD BE DISMISSED.

A. The record in this case is insufficient to permit consideration of the questions presented by the Government.

This Court is being asked by the Patent Office to rule on important factual issues concerning the technology of computers and computer programs. There is no record below that develops the nature of this technology. The Patent Office has not tendered to this Court or to the Court below any technological evidence supporting its position, nor could it seek to introduce any such evidence at this stage of the proceeding.[30] Since the record "is insufficient to permit decision of [the] claims,"[31] the writ of *certiorari* should be dismissed, *see Massachusetts v. Painten*, 389 U.S. 560.

The major technical fact issues raised by the legal contentions include:

a) Does programming a general-purpose digital computer restructure the computer into a new machine with a different structure? The Patent Board of Appeals, based on its technical expertise,[32] found that it does (Pet. 28A). The finding was not, nor could it have been, appealed to the C.C.P.A.[33] The Patent Office now asks this Court to overturn its own factual finding by arguing that the invention is not a machine structure but a "new use." CBEMA asserts that the invention is merely a "method of doing business." Neither the Patent Office nor CBEMA has offered one iota of evidence to support those arguments.

b) Are hardware- and software-built embodiments of the same machine technologically equivalent? The Patent Office's request for a ruling that a software built ma-

30. *See* 35 U.S.C. §144, which limits the C.C.P.A. to deciding the appeal "on the evidence produced before the Patent Office."

31. *Johnson v. Massachusetts*, 390 U.S. 511.

32. *See* 35 U.S.C. §7.

33. 35 U.S.C. §141, *et seq*.

chine system is unpatentable solely because it is built in software is at odds with the fact that special purpose computers can be built equivalently in hardware or software. The Patent Office relies only on contrary assertion and metaphor to attempt to justify different treatment under the Patent Code for the equivalent software-built machines, but can show no supporting evidence in the record.

c) Are the innovative aspects of computer programs generally and respondent's invention in particular machine structures? The Patent Office (*i.e.*, the Patent Board of Appeals) held that the invention is a special-purpose machine and a different machine from an unprogrammed general computer. The Patent Office and CBEMA now seek to offer an undocumented view of the programming technology which attempts to characterize programming (i) as an abstract activity unrelated to any specific machine structure and (ii) as a "routine" implementation of abstract ideas or a method of doing business. There are no facts in the record to support the assertions of the Patent Office or CBEMA.

This Court, unlike the two tribunals below, is not specially equipped, in the absence of a record, to deal with the intricate technological fact issues sought to be raised by the Patent Office.[34] This Court has recognized that in technical areas it should avoid disturbing the findings below, *see Brenner v. Manson*, 383 U.S. 519, 532. It is not appropriate to review technological issues without a record. Rather the Court should leave to other "parties whether

34. The Examiners-in-Chief of the U.S. Patent Office Board of Appeals are uniquely qualified by way of training in engineering and science to cope with the technology, *see* 35 U.S.C. §7 ". . . the Examiners-in-Chief shall be persons of competent legal knowledge and scientific ability." Likewise at the C.C.P.A., the judges have the assistance of specially qualified technical advisors with scientific ability. In addition, three of the judges are professionally trained in engineering and science having come out of a patent law background.

or not participants in [this appeal] a further opportunity to contest the matter on plenary records developed in litigation elsewhere. . . ." *Glidden v. Zdanok*, 370 U.S. 530, 579.

B. This Court is without jurisdiction to reverse findings of the Patent Board of Appeals favorable to respondent.

The Patent Office now asserts that programming a general-purpose computer does not restructure the machine but is merely a "new use" of the machine. This allegation has no basis, as the Board of Appeals held in a finding binding upon the Patent Office (Pet. 28A). This Court is without jurisdiction to reverse that finding.

The Board accepted restructuring as a technological fact beyond dispute, noting that the contrary view was "invalid." The Patent Office never sought to raise the restructuring issue in the C.C.P.A., nor by statute could the Patent Office appeal from that or any finding of its own Board of Appeals. 35 U.S.C. §141, *et seq*. The law is clear that only the applicant and not the Patent Office may appeal from the Board of Appeals to the C.C.P.A. and that the jurisdiction of the C.C.P.A. "extends to and embraces none other than the adverse decision from which appellant has [taken an] appeal." *In re Melott*, 331 F.2d 887.

28 U.S.C. §1256 limits the scope of this Court's appellate review to "cases in the C.C.P.A." That section must be read in light of the intent to bind the Patent Office implicit in the jurisdictional limitations of 35 U.S.C. §141, *et seq*.[35] Because the restructuring issue was not and could not have been in the C.C.P.A. case it cannot be considered by this Court.

The Patent Office attempt to raise its legal theory of "new use of an existing machine" also violates the legal

35. The Supreme Court has found it "appropriate that the Commissioner be bound by Board determinations." *Brenner v. Manson*, *supra*, at 523, fn. 6.

principle that no matter may be argued at the appellate court level which has not previously been raised before the lower court. That principle has particular applicability in this instance where the C.C.P.A.'s "special competence in complex, technical and important matters,"[36] would be indispensable.

Furthermore, the restructuring issue is not justiciable because the Patent Office is in reality seeking to appeal from itself. The Board of Appeals is the *alter ego* of the Commissioner of Patents and speaks for the Patent Office.[37] The Patent Office through the Board of Appeals has ruled that a programmed general-purpose computer is a different machine from an unprogrammed general-purpose computer. The Commissioner seeks to utilize the Supreme Court as an instrument for reversing his *alter ego* the Board of Appeals. Such intra-agency disagreements are not appropriately resolved by the judiciary. That is the reason the Commissioner by statute has no right of appeal from the Board of Appeals and why this Court must accept the determination of the Board on the restructuring issue.

C. *The basic question for which review was sought in this case—whether novel and unobvious machines constructed by programming a general-purpose digital computer are patentable—cannot be resolved by a decision in this case and Petitioner so concedes.*

This case was presented to the Court as a vehicle for deciding the broad issue of patentability of computer programs. That issue is the focus of both the Patent Office *certiorari* petition and its brief, which ask the Court to fashion the general rule that software-built special-

36. *Glidden Co. v. Zdanok*, 370 U.S. at 560.

37. See *Brenner v. Manson*, *supra*, at 523, fn. 6.

purpose computers are unpatentable subject matter.[38] This Court cannot issue such a rule of unpatentable subject matter under §101, based on the way a machine is built without violating the spirit and letter of the patent statute. Whether a particular machine constitutes patentable subject matter depends on the well-established principles of patent law—not whether the machine is built with one form of construction (software) as against another (hardware).

This Court cannot rule that all programs are unpatentable any more than it could rule a radio built with vacuum tubes is patentable, but if built with transistors, is unpatentable. The patent statute is broadly drafted, defining as patentable all machines regardless of the technology by which they are built. The only exception of a particular technological field, that of atomic weapons, is explicitly defined by Congress, 42 U.S.C. §281(a). Even the Patent Office in its reply brief concedes the "possibility of patentability" (R. Br. 2) of some programmed computers thereby negating any general ruling applicable to all programmed computers. Since all parties agree that there can be no answer to the broader question, the writ should be dismissed.[39]

The only other basis for decision offered by the Patent Office is the specific issue of the unpatentability of respondent's invention under 35 U.S.C. 103 for obviousness. Petitioner concedes this issue is not worthy of the Court's attention (Pet. 17, n. 10). The Patent Office does not urge that the incorrect legal standard was applied by the C.C.P.A. Instead the Patent Office merely repeats its

38. The Patent Office concedes that the remaining specific issue of the patentability of respondent's invention under 35 U.S.C. §103 is not certworthy, Petition for Certiorari 17, n. 10 (hereinafter referred to as "Pet."; the Brief for the Petitioner is referred to as "Br. Pet.").

39. The dismissal of *certiorari* would not leave the C.C.P.A. decision as a final adjudication of the patentability of respondent's invention. The validity of any patent that may issue to respondent may be fully attacked in an *inter partes* proceeding, (*see* Reviser's note to 35 U.S.C. §144) where a full record could be developed, *see Glidden*, *supra* at 578-79.

factual analysis of the content of the prior Dirks patent, which analysis was rejected by the CCPA. Thus the obviousness issue does not raise "an important question of federal law," (Sup. Ct. Rule 19) nor does the CCPA's mode of fact finding "conflict with applicable decisions of this Court," *id*. Unable to decide the general issue for which certiorari was sought and granted, this Court should exercise its discretion to decline to review the case and dismiss the writ of certiorari as improvidently granted.

II. SPECIAL-PURPOSE COMPUTERS BUILT BY PROGRAMMING A GENERAL-PURPOSE COMPUTER ARE MACHINES, HENCE PATENTABLE SUBJECT MATTER.

A. *The technology of computers and computer programs.*

A modern digital *computer* is an electronic machine that has the capacity to perform various data processing functions. The operation of a computer, like that of any automatic machine, is controlled by a program.

A *program* is a control mechanism which determines that each part of a particular machine system functions as required in order to achieve the desired result.[40]

A *fixed-purpose computer*, sometimes known as a "single-purpose computer," has one program physically built into its circuitry and thus can perform one and only one function. The program in such a machine is generally in its circuitry in the form of a configuration of sub-circuits or a predetermined sequence of switches or electronic impulses.

40. In earlier forms of automatic machines, programs "consist[ed] of mechanical parts such as gears, cams, and linkages that connect the various portions of a machine together and determine[d] the timing of the various actions of the machine." In modern computers, the program consists of the states of electronic circuits and "may be stored in the form of punched paper cards, paper tape, magnetic tape, ferrite cores, electronic circuits, or other mechanical or electronic memory devices." 2 *New Encyc. Britannica: Macropaedia*, 505, 506, "Automation"

A *general-purpose computer*, sometimes called a "programmable computer," can perform no function by itself without containing a program, as is true of any automatic machine.[41]

Both a general-purpose computer loaded with a particular program and a fixed-purpose computer are referred to as *special-purpose computers* because they can perform one particular function. The only difference is that the program in the fixed-purpose computer is permanently wired in (sometimes called "hardwired") while the program in a general-purpose computer is stored in the form of the states of electronic circuits (sometimes called "softwiring"). These two forms of construction are generally equivalent and functionally interchangeable.

The phrase "patents for computer programs" as used by the Patent Office to describe the key issue in this case, is technically imprecise and misleading in that what is sought to be protected in this technology is not some abstraction unrelated to a machine, but the structure of a particular special-purpose machine which may be produced by the softwired or programmed general-purpose computer but which also may be hardwired.[42]

The designer of a special-purpose computer is faced with the task of creating circuitry to achieve a particular goal or function. Prior to the advent of the stored program for a general-purpose computer, such complex circuitry had to be built by physically wiring the components. The great technological advantage of the stored program concept is that it permits the creation of circuits by the use of electronic signals without requiring externally made physical

41. *E.g.*, an automobile engine without camshaft or distributor cannot function.

42. "Program" has two meanings: one as used here is that control structure of a special-purpose computer (formed either by circuit states or configurations) which directs its operation as does the "program" of any automatic machine; the other common usage, not involved in this case, is a symbolic record of those signals in human-readable form, say a list of computer language statements.

connection. Each form of construction for special-purpose computers, softwiring or hardwiring, has its own technological and economic costs and benefits. The choice of which form to use in a given situation is a complex decision based on various engineering and management trade-offs.[43]

None of the factors concerning the choice of hard- or softwiring has any relation to inventiveness of the underlying machine. If the circuit states and configuration of a particular special-purpose computer contain the elements of invention, the machine is patentable and the fact that the wiring is "hard" or "soft" is immaterial. The interchangeability of these two types of computer construction is summed up in the phrase: "the engineering equivalence of software and hardware."

B. *A special-purpose computer built by programming a general-purpose computer is a different machine than the general-purpose computer with its own unique circuit structure.*

It is necessary to understand the manner in which programming does in fact restructure the general-purpose computer to appreciate the defect in the Government's argument that a programmed general-purpose computer is not a machine but merely the "new use of an existing machine." Far from being a mere semantic device or legal fiction, the restructuring concept is technological fact. When a program is loaded into a general-purpose computer, a set of circuit states and electric impulses are created which are the control or program of the special-purpose computer. It is because the general-purpose computer is programmable that it may appropriately be viewed as a "warehouse of parts."[44] Without a program, the function circuits of a general-purpose computer neither operate nor co-act in a "dynamic interrelationship,"[45] they are just like unconnected resistors and transistors on an engineer's workbench. Loading a program brings these function circuits into a synergistic and dynamic relation and structures them just like hardwiring the circuits of a fixed-purpose computer or wiring together resistors, capacitors and transistors to make a radio.

43. Frequently, the decision is not made until well along in the design process after the basic structure of the special-purpose computer has been determined.

44. Patent Office 1966 Guidelines, 829 *O.G.* 441, 442.

All the evidence supports the factual reality of the restructuring principle.[46] The Board of Appeals specifically accepted it in this case (Pet. 28A), and the C.C.P.A. has so found in a long line of decisions.[47] The Government by using distorted quotes, making unsupported and unsupportable assertions, and relying on false metaphors, has attempted to mislead the Court to believe the facts are otherwise.[48]

The Patent Office Brief presents two positions on the restructuring issue, the first is factual and the second seeks to oppose the first with a legal argument. First, the Patent Office concedes in its brief the facts of the restructuring principle, that without "appropriate controls" a general-purpose computer cannot perform any significant operations, and that

> [t]he particular operations to be performed by a computer are governed by the state and configuration of its electronic circuitry . . . . Such circuitry can be fully built into the computer by wires and other physi-

45. Br. Pet. 36.

46. For a collection of the statements of computer experts see Jacobs, "Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyrights and Patents," *Rutgers Journal of Computers and Law*, Vol. 1970, No. 8, at 50.

47. *In re Prater*, 415 F.2d 1393 (1969); *In re Bernhart*, 417 F.2d 1395 (1969); *In re Mahoney*, 421 F.2d 742 (1970); *In re Musgrave*, 431 F.2d 882 (1970); *In re Foster*, 438 F.2d 1011 (1971).

48. The Patent Office approach is necessary to its case in the sense that once the fact of restructuring is accepted, the bulk of the Patent Office case — the argument against the patentability of programmed computers — dissolves.

cal elements (*i.e.*, as part of the 'hardware'), in which case, unless it is physically rewired, the computer will ordinarily serve only a specific purpose or application . . . .

Alternatively, the computer may be designed so that the state or configuration or its circuitry can readily be changed without rewiring or other visible physical changes. Such changes are accomplished by the use of different 'programs'. . . . (Br. Pet. 14, 15)

In the above quote, the Patent Office concedes the entire basis of respondent's case by admitting, clearly and concisely, the interchangeability or equivalence of hardware and software.[49] The very same structural changes in the state and configuration of the general-purpose computer can be accomplished either by hardware or software. To base disparate legal treatment upon a mode of construction has no justification in law, technology or common sense. Softwired (*i.e.*, wireless) programming is an historic advance over the wired program. The patent statutes and their constitutional base were not intended to be limited to a finite period of technological development. There is no justification for the Patent Office's position that technology has now outstripped the boundaries of patent law.

Compelled by the facts to admit the interchangeability and equivalence of hardware and software and the basic restructuring principles of the stored program concept, the Patent Office presents a second and completely opposite brief seeking to obfuscate its admission and to divert the attention of the Court from the inescapable conclusion that software is, of course, patentable equally with the hardware. The Patent Office makes three arguments to deny restructuring:[50] 1. It quotes out of context Judge Rich's use of the phrase "legal fiction" as it pertains to the

49. CBEMA's Brief *amicus* avoids mention of the engineering equivalence of software and hardware though its members have used this principle for decades in the engineering of their hardware and software.

50. Br. Pet. 33-35.

legal effects of the drafting of patent claims, distorting his meaning and seeking to confuse this Court into believing that Judge Rich was casting doubt on the technological principles of restructuring.[51] 2. It distorts the Board's position.[52] If there is any one principle that both the Board of Appeals and the CCPA agreed on in this case, it is the fact of the restructuring principle, applicable to a stored-program computer. The Board of Appeals did not hesitate to remind the Examiner of the facts of computer technology that an unprogrammed or differently programmed computer is not the same machine as the one that results from programming a general-purpose computer (Pet. 28A.) 3. The Patent Office compares a player piano whose components are never restructured to a general-purpose computer whose usefulness is totally dependent upon the restructurability of its circuits (the ability of a stored program to connect its circuits into a desired configuration). The Patent Office's frequent use of this player piano analogy—not accepted by the Court in *Benson*[53]—is inexcusable since it ignores the unique quality of general-purpose computers: without a program they have no function but they can be programmed to connect the elementary circuits of the general-purpose computer into an unlimited variety of new configurations, each generally unrelated to the elementary circuit functions. Only the general-purpose computer, not the player piano, is restructured by a program into a different machine, with a different function.

51. In fact, Judge Rich has made it clear in his "dissent" and in a subsequent writing (*see* fn. 28 *supra*) that he has not retreated from his agreement with the C.C.P.A.'s factual findings on restructuring set forth in *Prater* and *Bernhart*.

52. The Patent Office quotes the Board's opinion dealing with language in the patent claims that the Board questioned on formal grounds under 35 U.S.C. §112, and uses those quotes to argue "new use," on which the Board had reversed the Examiner.

53. *Gottschalk v. Benson*, 409 U.S. 63.

III. SOFTWARE IS NOT AN "ABSTRACT IDEA."

A. *The engineering of software.*

The software products segment of the computer industry is composed of hundreds of small and medium size companies, many of which are members of ADAPSO/SIA. On the staff of these companies are thousands of personnel actively engaged in programming. These programmers have I.Q.'s "well above the average college graduate."[54] Many have degrees in computer sciences, not infrequently at the masters and doctorate level. Many work with the precision of engineers in a field increasingly known as software engineering. They build software out of building blocks[55] to engineer the special-purpose computer in a hierarchical structure,[56] much the same as that required for other modern complex machines.

The automobile, for example, is a hierarchical machine system. The designer of the gearbox combines gears and other mechanical parts in a particular way without having to be concerned with or having any knowledge of the molecular structure of the steel used in the gears (which concern is left to a metallurgist). The designer of the overall transmission system simply configures gears, drive shaft, universal, and other elements without concern for the lower hierarchical levels of the specific internal structure of those individual components or of the structure of the steel.

In the hierarchy of computer systems, the logic circuit designer builds logic circuits such as "AND" and "OR" circuits without concern for the molecular structure of the individual transistors and other components that make up

54. *Psychology of Programming*, G. Weinberg, Van Nostrand Press, p. 161.

55. The building blocks of software and hardware are discussed in Jacobs, *supra*.

56. See "Hierarchies—Software Engineering Design," *Software Engineering, International Computer State of the Art Report*, 1972, p. 47 *et seq*.

those electronic circuits (which concern is left to component designers working at a lower level of hierarchy). The designer of the hardware function circuits such as "ADD," "COMPARE" and "STORE," configures logic circuits without concern for the detailed structure of the individual logic circuits he uses to construct the hardware function circuits. Similarly, the "programmer," the designer of the special-purpose computer, configures hardware function circuits into the structure of the ultimate special-purpose machine without concern for the detailed structure of the individual hardware function circuits. The same hierarchical process is followed if, instead of softwire programming, hardwire programming is used to build the special-purpose computer.

These building blocks lie on successive levels of a hierarchy[57] of circuit construction and the software engineering is performed at the highest level. The software itself, because of its complexity, often must also be constructed in a number of hierarchical levels, and the software engineer should preferably work like:

> ". . . the electrical engineer designing a new T.V. set [who] certainly doesn't design each vacuum tube, transistor, and capacitor all over again; he relies on existing components. In fact, he normally designs on a much higher level, using 'off-the-shelf' power supplies, oscillators, etc."[58]

A recent IBM report further demonstrates the similarities in software and hardware engineering:

> Structured programming reduces the arrangement of the program logic to a process like that found in [hardware] engineering where logic circuits are constructed from a basic set of figures. As such, it

57. See Jacobs, *supra*, for a discussion of the circuit interrelationships in the different hierarchical levels of a programmed computer.

58. The IBM report, "Composite Design: The Design of Modular Programs," Jan. 1973, p. 6.

represents a standard based on a solid theoretical foundation.[59]

Working at different levels in the system hierarchy of the special-purpose computer to be built, contemporary software engineers as that IBM report explains, may use a "structured programming"[60] approach similar to the design of other automatic machines:

> The top-down approach is patterned after the natural approach to system design and requires that programming proceed from developing the control architecture (interface) statements and initial data definitions downward to developing and integrating the functional units. Top-down programming is an ordering of system development which allows for continual integration of the system parts as they are developed and provides for interfaces prior to the parts being developed.[61]

In the Apollo Space Shuttle Software System, a mammoth software development effort:

> . . . program structuring is of major importance; decisions . . . determine the total structure of the software both statically and dynamically. Program structuring defines the building blocks, the control mechanisms and the interfaces of the software. It is in this effort that the interrelationship of systems software, applications software and data is entirely mapped out.[62]

The software engineer builds complex automatic machines as the foregoing examples amply demonstrates. Neither the software engineer nor the hardware engineer

59. "Management Concepts for Top-Down Structured Programming," IBM Report, February 1973, p. 1.

60. *Ibid.*, Abstract.

61. *Ibid.*, p. 6.

62. "Top-Down, Bottom-Up, Structured Programming & Program Structure," Draper Labs, MIT, Dec., 1972, pp. 28, 31.

devises "abstract ideas," or methods of transacting business," contrary to the Patent Office's suggestions (Br. Pet. 21), nor a "mathematical formula, a method of doing business, or an expression of a scientific principle or law of nature," as CBEMA would have the Court believe (Br. CBEMA 5). The software engineer is not constructing "new uses of an existing machine" nor anything resembling a "piano roll [in] an old player piano" (Br. Pet. 35).

Relying on the restructuring principle, the Board found that the respondent's program restructures the general purpose computer into a different machine and not a "new use." This issue did not arise in *Benson*, which dealt with process claims for a "mathematical formula," not with machines. Certainly the Court in Benson did not conclude that this technology was to be excluded from the patent system.

B. *The invention process in software.*

The design and development process in this technology is quite similar to that of any other technology. First the goal is decided upon, then the prior art is explored for possible solutions and, if not found, the "technological roadblocks" are identified.[63] Invention may take place in overcoming these roadblocks.

Parallels to the design and development process of other machines may be seen in the software design process presented in a paper at the World NATO Conference entitled "Software Engineering"[64] which took place in Germany in 1969 and was attended by over 50 experts and prominent leaders in the software field. The first seven stages[65] of this process from the concep-

63. *See, e.g., Computers in the 1980's*, R. Turn, Columbia Univ. Press, 1974, p. 21.

64. "Software Engineering," Report on a World Conference Sponsored by the NATO Science Committee, Garmisch, Germany, Jan., 1969, p. 13.

65. "a. Specification of the complete hardware-software system. b. Definition of the functions to be performed by the pro-

tion "of the complete hardware-software system" of the special-purpose computer to be built until the "detail design" is worked out, all involve areas for overcoming "technological roadblocks" and solving machine problems. Invention in the ultimate special-purpose computer may result from the working at the various hierarchical levels of this design process. An additional six stages in the above-cited design process involve testing and operation of the overall machine as well as its parts to determine if the "software-hardware package meets all of its design requirements." If not, further work and debugging at various of the first seven stages may result in other opportunity for invention.

Nothing in this design process is a "routine" task, as the Patent Office misleadingly suggests.[66] The design process suggested by the Patent Office (Br. Pet. 16-17) and CBEMA (Br. 5-6) bears no relation to that of the real computer world.

IV. COPYRIGHTS CANNOT ~~PREVENT~~ PROTECT MACHINE INVENTIONS; PATENTS, WHICH CAN, ARE NEEDED FOR SOFTWARE.

CBEMA's suggestions that copyright is a viable method of protecting inventions in programmed com-

Note 65—*Continued*
gram. c. Design and documentation of the master overall program plan. d. Subdivision of the large system program into manageable program blocks. e. At this point, the interfaces between the program blocks must be precisely defined and documented. Since the usual means of passing data between program jobs is via the use of data in the memory of the system, the fundamental program progress data in memory should be formulated and defined for the interfaces between each program block. f. Basic program subroutines defined and documented. g. Detail design, coding and documentation of each program block."

66. Unlike the creative process of designing the special purpose computer, the resulting machine is designed so that its mode of operation is "routine." A complex control task is performed by a combination of elementary machine circuits that are operated over and over again in the same exact way.

puters is contradicted by the constitutional limitation of copyright protection to "writings." Copyrights may only be used to protect particular printed records and descriptions of machines, but not the actual machine structure that they represent. To one whose innovation is a machine structure for a special-purpose computer built with software, only patent protection will suffice.

The copyright laws contain no novelty requirement; exclusive rights are predicated minimally on originality. CBEMA favors copyright protection because its members apparently are not interested in protecting the innovation in software (which may exist in 1% of all programs) but are interested in protecting the non-innovative aspects of software (perhaps 99% of all programs), the part of a machine which has never been protectable, neither by patents nor by copyrights. Such use of the copyright laws to protect non-inventive forms of programmed machines would be a circumvention of the constitutional requirement of novelty (inventiveness) for obtaining exclusive rights in a machine structure. *Graham v. John Deere Co.*, 383 U.S. 1.

Proposals have been made for registration systems (either similar to that of copyrights or of other forms) for the registration of computer programs. A registration system in the U.S. would have the same fundamental objections as those for copyright protection, because it would provide protection in the nature of exclusive rights for machine technology without being based on the constitutional requirements of novelty and invention.

The World Intellectual Property Organization, in considering a registration system that might be suitable in European and other countries, indicated that such a registration system should not be in lieu of patent protection for this technology, but in addition thereto. WIPO made it clear in 1974 that "patent protection should not be withheld from inventions for the reason that they involve computer programs,"[67] and again in 1975

67. WIPO-AGCP/NGO/9, July 5, 1974, p. 8, ¶ 58(b).

because they "involve a software element."[68] Thus WIPO indicates that it makes no sense to set a basis for patentability solely on the grounds of the mode of construction of the machine—the entire basis of the Patent Office's case.

Software is a new technology that is today viewed as the most critical and important element of a computer system. As in any new technology, for it to grow and compete, it needs all the protection due it under the law. ADAPSO/SIA asks the Court to treat the software technology equal to the hardware technology—no better—and no worse.

CONCLUSION

For the reasons stated above, it is respectfully urged that the writ of *certiorari* be dismissed. In the alternative, if this Court reaches the merits of the case, the decision of the Court of Customs and Patent Appeals should be affirmed.

Respectfully submitted,

DAVID COHEN
Attorney for Amicus
ADAPSO/SIA

68. WIPO-AGCP/NGO/II/11, July 14, 1975, p. 3.